

K-1. dcextract

1.0 General Information

1.1 Application Description

The purpose of the dcextract program is to pull data from the processed SHEF data value tables of the RFC Archive database and write it to a file in DATACARD format. It also creates a columnar text file for importing to spreadsheets. This program is written in C.

1.2 Design Considerations

The program writes some general comment information about where it found the data in the database at the top of the file. This includes the database table, as well as the lid and SHEF code it was stored under. The body of the file contains four values per line.

If the data stored in the database is not continuous (there are no rows stored for some dates), the program will put missing values into the datacard output for these time periods. Also, the program will make sure the data written to the file starts on the first of the month. If the first available data is before the 15th, missing values will be written to fill in the first part of the month. If the first available data is after the 15th, the datacard output will not start until the next month.

1.3 Application Assumptions

The archive database data tables store all data in Zulu (Z) time, while the DATACARD format is written in local time. To convert from Z time to local time, dcextract assumes that local time is 4-8 hours different from Z time. The same assumptions are made (in reverse) in the dcparse program, which SHEF encodes data from the datacard format. Here is how the conversion is handled in the program:

- Daily Data
 - The *peDPsep* (daily data table) data slots have an end time of 12z for the given day.
 - Day 1 of the database is written to day 1 of the DATACARD; exception:
 - If the `--shift` command line argument is used day 2 of the database is written to day 1 of the DATACARD.
- Six Hourly Data
 - The *peQPsep* (six hourly data table) data slots have ending times of 0z, 6z, 12z, and 18z for the given day.
 - First value in DATACARD is midnight to 6 am local time on day 1, meaning that it ends between 10z and 14z on database day 1.

- This program writes the database 12z value as the DATACARD 6 AM value.
- Hourly Data
 - The *peHPsep* (hourly data table) data slots are 0z, 1z, 2z, etc. for the given day.
 - First value in DATACARD is 1 am local time on day 1.
 - This program asks the user to input the number of hours local time is offset from Z time. For example, if local time is 6 hours different from Z time, the DATACARD 1 AM value is written from the database 7z value.

1.4 Enhancements/Bug Fixes/Changes

Build OB7.2

This application was updated to take into account the upgrade of the OS in ob7.2 and the change in RDBMS from Informix to Postgres. Documentation was updated to reflect these changes.

Build OB6

Changes

- Fixes ADB Bug R1-50. No longer automatically shifts the database values back one day before writing to the DATACARD. The original program made the following assumption for daily data:
 - First value in DATACARD is midnight to midnight local time on day 1, meaning that it ends between 4z and 8z on database day 2.
- Added a command line argument (-shift) to force the 'day shift' for those who want it.
- Fixes ADB Bug R1-22. Fills in with missing data when rows are not found in the database so the record is continuous.

Build OB5

Enhancements

- Add 'debug' mode (-d command line argument) and more error and warning messages. (ADB ER-8)
- Add apps_defaults token 'dcextract_outdir' for location of output file. (ADB ER-8)
- Make sure hourly and six-hourly data begin on the first day of the month.
- Fix problem printing years greater than '99' with just one number.

2.0 Configuration Information

The following apps_defaults tokens are used:

| | |
|------------------|--|
| adb_name | name of the archive database |
| dcextract_outdir | output directory for datacard files (optional) |

The DATACARD output files are created with a name of *lid.shefcode* and the columnar output file with a name of *lid.shefcode.col*. If the `dcextract_outdir` token is defined the output files will be created in that directory, otherwise they will be in the users current working directory.

3.0 User How-To

This program can be run either through the *arcmenu* -> *Other Viewing/Extraction menu*, or on the command line by simply typing *dcextract*. The program requires some user input, which can either be included on the command line or input as prompted by the program. The required inputs are as follows:

1. location identifier (lid) for data to extract
2. SHEF parameter code for data to extract
3. first year of data to extract (default is first year of stored record)
4. last year of data to extract (default is last year of stored record)
5. number of hours local time is offset from Z time (needed only for hourly data)

There is an optional command line argument that controls the 'day shift' (this must be entered before any other arguments if it is used):

-shift = write database day 2 to DATACARD day 1

There is an optional command line argument that controls the 'debug' mode (this must be entered before any other arguments if it is used):

-d = debug mode; writes more program information to the screen

Examples of running program using command line arguments:

dcextract -d alec2 ppq5zzz (debug mode on, uses default first and last years)

dcextract gjt pph5zzz 1950 2000 6 (hourly data needs offset)

dcextract -shift alec2 qrd5zzz (daily data with day shift)

If no command line arguments are given, the user will be prompted for the required information.

4.0 Troubleshooting Information

Use the 'debug' mode to see the database query being used and when missing data is being filled. If the user still has problems, contact the RFC Support Group for assistance.